Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2001 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2001

Requirements Engineering for Enterprise Systems

Michael Rosemann
Queensland University of Technology

REQUIREMENTS ENGINEERING FOR ENTERPRISE SYSTEMS

Michael Rosemann
Queensland University of Technology
m.rosemann@qut.edu.au

Abstract

Requirements Engineering for Enterprise Systems can be differentiated in developing and configuring these comprehensive solutions. This paper focuses on the system configuration. This process is supported by implementation tools, which guide the project team through the system individualization. A main activity within this process is the configuration of reference models, which support the specification of requirements related to Enterprise Systems. Enterprise Systems reference models, however, do not depict the actual configuration potential of the system, but rather describe the complete system functionality and structure. This paper suggests modifications of reference modeling techniques for Enterprise Systems. These extensions allow the exact specification of configuration opportunities. Thus, they facilitate the use of conceptual models for all stages of an Enterprise System lifecycle. The presented research uses reference data models and the market leading mySAP applications as examples.

Enterprise System Lifecycle

The process of introducing Enterprise Systems is significantly different from the well-discussed classical software engineering lifecycle (Figure 1). After a common phase of defining project objectives, analyzing the current situation and developing a business blueprint (to-be-model), Enterprise Systems management contains the two main phases of system selection and configuration (Kirchmer, 2000). *Configuration management* includes the individualization of the system and the development of add-on functionality. This activity if of enormous complexity as the configuration of an Enterprise Systems requires the comprehensive selection of relevant parameters. This process can easily contain the evaluation of more than 1,000 alternatives.

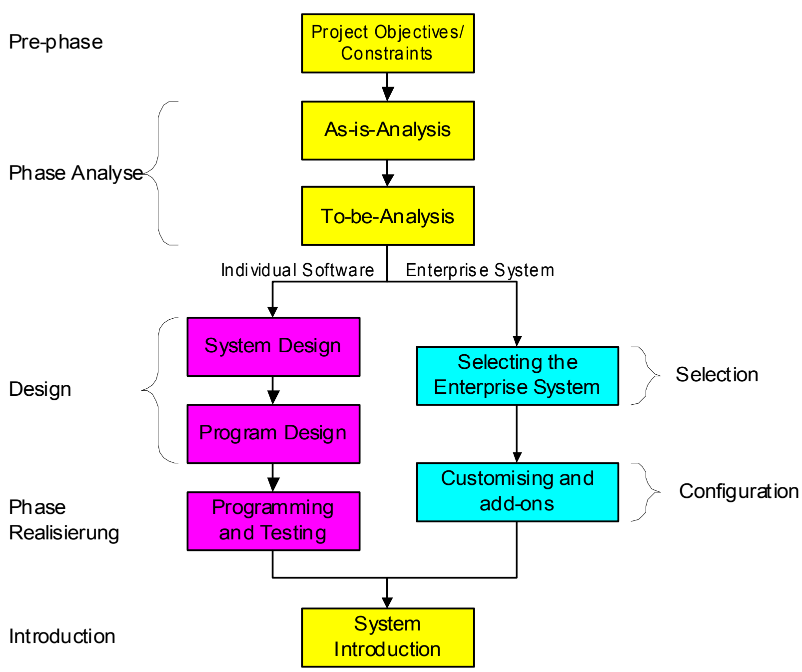

Figure 1. Differences in the System Lifecycles

Most Enterprise Systems support the implementation team with step-by-step guidelines, which lead the project team through the required configuration process. Similar to the classical software development process, the configuration of Enterprise Systems has a requirements engineering phase. However, instead of developing conceptual models from scratch, requirements engineering for Enterprise Systems consists of a continuous process of specifying reference models. This specification includes the selection between given alternatives, the evaluation of optional alternatives, the confirmation and eventual modification of defaults, the entering of master data for the organizational structure, and the assignment of data. Consequently, requirements engineering for Enterprise Systems starts with the "big picture", which is reduced to the relevant part.

Enterprise Systems reference models have been developed as a conceptual level on top of these hands-on configurations. They include in the most cases at least reference solutions for the organizational structure, the data and the supported business processes. These reference models are significantly different from software-independent reference models that describe recommended solutions from the viewpoint of academics (e.g. Scheer, 1998) of for consulting purposes. In contrast, Enterprise Systems reference models are conceptual descriptions of customizable software. It depends on the functionality of the software to what extent they can be regarded as benchmarks.

Though most of the Fortune 1000 companies currently use Enterprise Systems, IS literature has ignored the conceptual problems related to requirements engineering for Enterprise Systems. This area of research can be divided into requirements engineering for the *development of Enterprise Systems* (Brinkkemper, 1999; Daneva, 2000) and requirements engineering for the *configuration of Enterprise Systems*. The latter one is the focus of this paper. Theoretical contributions in this field are still the exception. As an example, Rolland and Prakesh suggest a map including ERP goals and objectives for the identification and evaluation of user needs. Gulla and Brasethvik (2000) suggest three process modelling tiers to manage the complexity of process modelling in comprehensive Enterprise Systems projects. Their functional tier dimension deals with the functionality of the Enterprise System. However, they do not discuss how to differentiate models in this tier.

A similar status can be observed in practice, where even the most popular solutions for Enterprise Systems (mySAP applications) still use classical modelling techniques (extended ER-models and Event-driven process chains) (Curran and Keller, 1998). These techniques do not take the special requirements of configuration management into account.

This paper proposes how current reference modeling techniques could be extended for the purpose of configuration management. As an example, it discusses the configuration of reference data models. This paper classifies possible configurations of data models and suggests ways how to model these configurations. Corresponding suggestions for the configuration of reference process models have been suggested (Rosemann, 2000).

Enterprise Systems Reference Models

Enterprise Systems (ES) are highly integrated business solutions for typical functional areas such as procurement, material management, production, sales and distribution, financial accounting, and human resource management. These functions are often further individualized for several industries such as Automotive, Retailing, High Tech, etc. Consequently, Enterprise Systems tend to be very comprehensive and complex as they have to combine the different interests of various customer groups, countries, and industries.

As one approach to improve the understandability of Enterprise Systems, ES vendors developed since 1992 reference models. They exist in the form of function, data, system organization, object and process models and support different levels of abstraction. It is usually possible to refer from the models to the relevant part of the online documentation and on the lowest level of abstraction even to corresponding transactions. It has to be stressed that these models are also designed for the end users of Enterprise Systems, and not only for the implementation team. They specify how processes can be executed by the end users. Consequently, end users benefit from these models as they comprehensively and quickly inform about the related software functionality using business terms. These reference models are part of most ES solution and do not have to be purchased separately.

Though these models contributed significantly to the understandability of Enterprise Systems' functionality, they still have core weaknesses.

- As the models are focused on the description of the process execution and the data structure, it is not obvious what *configuration alternatives* exist. The analysis of a reference model shows what is possible in general, but not what might be recommended alternatives. They represent the entire functionality from the viewpoint that the complete system is used. The models are not designed for configuration. Their modelling techniques do not support constructs that cover possible decisions during the implementation phase, i.e. decisions at buildtime. Thus, they do not differentiate between decisions on instance level and type level.
- Reference models concentrate on the elements that are of importance for a specific Enterprise System. *Enterprise-individual aspects* of the organization, business objectives or manual tasks cannot be seen in these models. They do not include any references to the involved or required knowledge.

- Besides the missing transparency regarding possible choices during the configuration process, it is also not clear what *consequences* a configuration of one process or data structure has *on other processes or data structure*. An example for the interrelation between the configuration of Enterprise Systems processes can be found in Rosemann (2000).
- Moreover, the models do not have any *link to the actual process execution or database design*. Thus, it is not possible (e.g. in the form of model attributes) to see the process performance expressed in key performance indicators such as processing time or resource utilzation).

Overall, reference models are a comprehensive and consolidated description of possible processes and data of an Enterprise System. This paper discusses how the meta models of reference models can be extended in order to depict configuration opportunities. The assumption is that this will increase the value of these models and their acceptance. he following discussions will use reference data models as an example.

Configuration of Reference Data Models

Reference data models summarize on a conceptual level the data structure of an Enterprise System. They are typically based on modeling techniques that extend the classical Entity-Relationship approach suggested by Chen (1976). An extension is, among others, the generalization-specialization relationship type. One of the most comprehensive reference data models is the SAP reference model, which includes more than 4,000 entity types. More than 180 business objects cluster this data model (Curran and Keller, 1998).

A main application area for reference data models are decisions about system organizational units such as legal entity, purchasing organization or distribution channel. Data models depict in these cases precisely the given opportunities of an Enterprise System. A small subset of the entire reference data model (approx. 30-40 entity types) allows a complete description of the interrelations between system organizational units. This facilitates especially configuration decisions that cover more than one module. Furthermore, reference data models are used within the development of add-on solutions for Enterprise Systems.

The following paragraphs discuss main configuration decisions and how they could be depicted in reference data models. Extracts of the SAP reference data model are used as examples. However, the recommended suggestions are independent from the actual ER-dialect and can be applied to different ER-approaches as well.

Optional Entity Types

Many examples for model configurations related to optional entity types can be found in Enterprise Systems in the definition of the organizational structure. The Financial Accounting solution in SAP R/3, for example, requires a decision about the optional system organizational unit ‘Business Area’. Business Areas in SAP R/3 are defined as "the organisational unit in external accounting that corresponds to a selected area of activity or responsibility within an organization to which the value movements entered in financial accounting can be assigned." (SAP online documentation, 2000).

Figure 2. Optional Entity Type "Business Area" with a Required Decision

Figure 2 shows the two entity types Company Code and Business Area. A Company Code in SAP represents a legal entity with own balance sheets and profit and loss statements. Though this Figure is simplified regarding the cardinalities, it is obvious, that the Business Area existentially depends on the Company Code. In order to highlight the optional nature of a Business Area and thus the required configuration, a dotted line is suggested for optional entity types. The decision, however, about this organizational unit is compulsory as other processes depend on this configuration. In SAP, the process of entering a Cost Center (controlling process) requires a reference to a Business Area for each Cost Center, if Business Areas are used.

This simple example already highlights the effect of changes to a reference data model. Required configurations become obvious and can be studied in the context of semantic relationships depicted in the reference data model. Besides the graphical change of the notation for the entity type, further information could support the related configuration. Examples would be a reference to the relevant online help or a comparison with alternative constructs in the system (e.g. organizational units in controlling). If available, even benchmarks from the industry about this configuration would be possible. This would answer questions such as "In what situations are companies using Business Areas?" This information could also provide valuable information about decisions on the instance level, for example, "How many Business Areas are recommended and for what purposes?"

Other important examples for such optional entity types are specializations of organizational units. During the system configuration process, the relevant specializations of an organizational unit have to be selected. Most specializations are typical optional.

Other organizational units are optional *and* the entire configuration process does not necessarily require a decision about these constructs. In these cases, the system typically sets one instance of this organizational unit as a (hidden) default. If a decision is made to use many instances of this organizational unit, an entity type in the corresponding data model is required. An example in SAP R/3 is the Dunning Area. A Dunning Area groups in SAP Financial Accounting-Accounts Payable all customers that are treated in the same way when in comes to sending out reminder notices. No other process outside the dunning sub-module depends on this decision. All customers are treated identically, if Dunning Areas are not used. It is proposed to highlight optional entity types that do not require a decision during the system individualization with two dotted lines (Figure 3).

Figure 3. Optional Entity Type "Dunning Area" with an Optional Decision

Configuration of Relationship Types

The configuration of relationship types includes two decisions. First, if the relationship type at all is required. Second, what cardinalities the relationship should have. Optional relationships in Enterprise Systems support especially cross-references. The more cross-references exist, the more linkages are established between different modules and sub-modules. On the other hand, intensive cross-referencing demands a good understanding of these interdependencies from the system users and can be perceived as restrictive in operational processes (e.g., many cost objects in posting transactions).

Optional Relationship Types

Consequently, many examples for optional relationship types can be found in the accounting modules of Enterprise Systems. An example is given in Figure 4, which shows the interrelation between Profit Center and Cost Center in SAP R/3. The arrows indicate that a 0,m-1,1 relationship type exists, i.e. if a customizing decision for this relationship is made, each Cost Center has to refer to exactly one Profit Center. Each Profit Center consolidates the cost of zero to many Cost Centers. The dotted line indicates that this is an optional relationship type. In comparison, the actual SAP reference data model characterizes this relationship only as conditional. It does not become clear, if the decision to link a Cost Center to a Profit Center can be made at buildtime for all Cost Centers or at runtime for each new Cost Center separately.

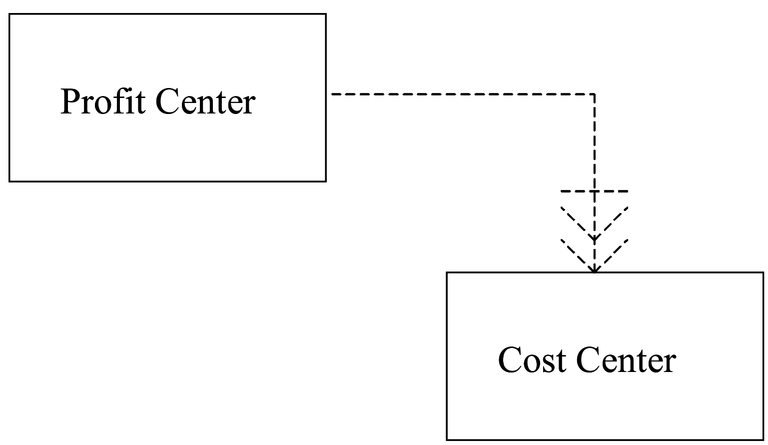

Figure 4. Optional Relationship Type

Further model extensions could be made that differentiate the default value for this relationship type in the case that no decision is made during the implementation process. Corresponding to the suggestions regarding optional entity types, optional relationship types that do not require a decision during the configuration process could be depicted with two dotted lines.

Configuration of Cardinalities

Decisions about the configuration of cardinalities can take place in conjunction with optional relationship types or independent from them. An example related to Figure 4 would be that after a (buildtime) decision to establish the relationship type between Cost Center and Profit Center, a decision has to be made, if an individual Cost Center definitely has to refer to a profit center or not. This difference can be expressed by the minimum cardinalities 0 and 1 and expresses a possible decision at runtime. In general, many decisions during the configuration of an Enterprise System can be related to minimum and maximum cardinalities. Unlike optional entity types and optional relationship types, cardinalities that are not predefined have to be configured during the customizing process.

Regarding the minimum cardinalities, a decision has to be made between 0 and 1, whereas 0 indicates that at runtime an entity of the involved entity type does not have to take part in the relationship. The typical alternative for the maximum cardinality is 1 and many. An example for such a configuration alternative is given in Figure 5. This example from SAP R/3 shows the interrelation between Company Code, the highest reporting unit in SAP Financial Accounting, and Controlling Area, the highest reporting unit in Cost Management. A variable is used as a maximum cardinality in order to express that a Controlling Area either corresponds with exactly one Company Code (x = 1) or it covers more than one Company Code (x = many). This is a mandatory decision during the configuration of SAP's accounting solution. Again, the existing SAP data model only includes the maximum case, i.e. x = many. The actual configuration opportunity does not become obvious.

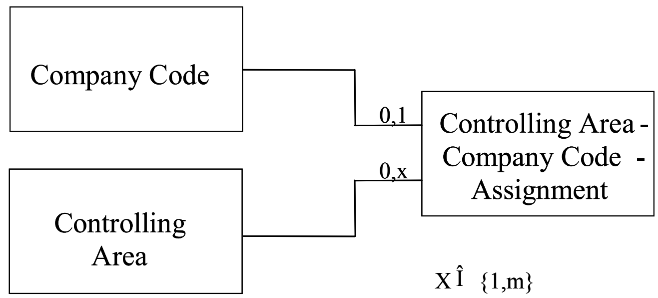

Figure 5. Configuration of Cardinalities

Interrelations Between Configurations

All the configurations above were local customizing decisions. Each configuration could be made in the local context of the involved entity types, relationship types and cardinalities. Further complexity is added, if also interrelated configurations are analysed. These interrelations can be differentiated again in mandatory and optional interrelationships. In the case of optional interrelations, a pointer should at least refer to related configurations. Table 1 describes the possible 9 alternative interrelations, if the configuration of entity types, relationship types and cardinalities is considered. This table can be interpreted as follows: Case 1 describes the situation, in which a decision about one entity type influences the decision about another entity type.

Table 1. Possible Interrelations Between Configurations

	Entity Types	Relationship Type	Cardinality
Entity Type	case 1	case 2	case 3
Relationship Type	case 4	case 5	case 6
Cardinality	case 7	case 8	case 9

These interrelations require a combination of te suggested modifications of existing reference modeling techniques. Pointers would be useful in these cases in order to indicate existing dependencies. Selected examples in the SAP R/3 system are the following.

- The decision to use Profit Centers in the Cost Management module can motivate not to use Business Areas in Financial Accounting as Profit Centers have comprehensive reporting opportunities (optional interrelation) (case 1).
- Company Codes in Financial Accounting and Purchasing Organizations as the highest reporting units in Material Management can be linked in SAP R/3 either directly or via individual Plants (case 5).
- A key configuration regarding the linkage of Materials Management and Financial Accounting has to be made regarding the level, on which material should be evaluated. This can be either the level of the legal entity (Company Code) or each individual factory (Plant). Thus, the decisions about the Company Code-Material Valuation Area relationship type and about the Material Valuation Area-Plant relationship type have to be made simultaneously (case 9). One of these cardinalities will be 1-1, the other one 1-many.

Limitations and Outlook

This paper has been focused on reference *data* models. First suggestions for the configuration of reference *process* models have already been proposed in Rosemann (2000). At this stage, this entire research is concentrated on vendor-specific methodologies as it uses reference process and data models in mySAP applications. This represents a major *limitation* of this work, though the general recommendations can easily be adapted to techniques used by other Enterprise Systems vendors.

Future research will be focused on the following tasks:

- Analysis of the completeness of the suggested configuration opportunities. Using a main sub-module in SAP R/3 (e.g. configuration of the enterprise structure or Accounts Payable), the consequences of the required system configurations on the reference data models will be studied.
- Current research analyzed the configuration of reference data models and reference process models separately. Future work will investigate in linking both configurations and increasing the consistency and integrity of model-based configurations.
- The acceptance and usefulness of extended reference models will be studied with internal and external project members of current Enterprise Systems implementation and upgrade projects. These empirical results might motivate changes or further extensions of the suggested model-based configurations.
- A long-term goal of this work in the integration of benchmarking data into the reference models (e.g. typical configurations in a certain industry). This will support configuration decisions grounded on the experiences of certain industries.

In a connected area of research, we also aim for an ontological analysis of our extended meta model for Enterprise Systems reference models (Weber 1997; Rosemann and Green, 2000).

References

Brinkkemper, S. (1999). Requirements Engineering for ERP: Requirements Management for the Development of Packaged Software. Proceedings of the 4th International Symposium on Requirements Engineering. Limerick, Ireland, 7-11 June.

Chen , P P-S. (1976). Entity-Relationship Model: Towards a Unified View of Data. ACM Transactions on Database Systems. Vol. 1, No. 1, pp. 9-36.

Curran, Th., and G. Keller (1998). SAP R/3 Business Blueprint. Prentice Hall: Upper Saddle River.

Daneva, M. (2000) An Assessment of the Effects of Requirements Reuse Measurements on the ERP Requirements Engineering Process. Proceedings of the 4th International Conference on Requirements Engineering. Schaumburg, Ill., 19-23 June.

Gulla, J. A., and T. Brasethvik (2000). On the Challenges of Business Modeling in Large Scale Reengineering Projects. Proceedings of the 4th International Conference on Requirements Engineering. Schaumburg, Ill., 19-23 June, pp. 17-26.

Kirchmer, M. (2000). Business Process oriented Implementation of Standard Software. 2 nd ed., Springer-Verlag: Berlin et al.

Rolland, C., and N. Prakash (2000). Bridging the Gap between Organisational Needs and ERP Functionality. Requirements Engineering, Vol. 5, Issues 3, pp. 180-193.

Rosemann, M. (2000) Using Reference Models within the Enterprise Resource Planning Lifecycle. Australian Accounting Review, Vol. 11, No. 2, pp. 19-31.

Rosemann, M. and P. Green (2000). Integrating Multi-Perspectives into Ontologies. Proceedings of the International Conference on Information Systems – ICIS 2000. Eds.: W. J. Orlikowski et al., Brisbane, 10-13 December.

Scheer, A-W. (1998). ARIS - Business Process Modeling, 3rd ed., Springer-Verlag: Berlin et al.

Weber, R. (1997). Ontological Foundations of Information Systems. Melbourne.